2020-05-23 - Handout – Dynamic Programming

# Q1. 0-1 Knapsack problems (Bounded knapsack)

Link: <https://www.geeksforgeeks.org/0-1-knapsack-problem-dp-10/>

Given weights and values of n items, put these items in a knapsack of capacity W to get the maximum total value in the knapsack. In other words, given two integer arrays val[0..n-1] and wt[0..n-1] which represent values and weights associated with n items respectively. Also given an integer W which represents knapsack capacity, find out the maximum value subset of val[] such that sum of the weights of this subset is smaller than or equal to W. You cannot break an item, either pick the complete item or don’t pick it (0-1 property).

# 

# Q2. Subset Sum

Link: <https://www.geeksforgeeks.org/subset-sum-problem-dp-25/?ref=rp>

Given a set of non-negative integers, and a value *sum*, determine if there is a subset of the given set with sum equal to given *sum*.

|  |  |
| --- | --- |
| **Example 1: Input: set[] = {3, 34, 4, 12, 5, 2}, sum = 9 Output: True  There is a subset (4, 5) with sum 9.** | **Example 2: Input: set[] = {3, 34, 4, 12, 5, 2}, sum = 30 Output: False There is no subset that add up to 30.** |

# Q3. Equal Subset sum

Link: <https://leetcode.com/problems/partition-equal-subset-sum/>

Given a non-empty array containing only positive integers, find if the array can be partitioned into two subsets such that the sum of elements in both subsets is equal.

Note:

* Each of the array element will not exceed 100.
* The array size will not exceed 200.

|  |  |
| --- | --- |
| **Example 1: Input: [1, 5, 11, 5] Output: true Explanation: The array can be partitioned as [1, 5, 5] and [11].** | **Example 2: Input: [1, 2, 3, 5] Output: false Explanation: The array cannot be partitioned into equal sum subsets.** |